

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The version of the following full text has not yet been defined or was untraceable and may differ from the publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/74347>

Please be advised that this information was generated on 2017-12-06 and may be subject to change.

Parallels and dichotomies in Interface Design & Document Design

Piet Westendorp[@]

Carel Jansen[@]

Rob Punselie[@]

If you can't explain it, don't design it.

Summary

The design of the user interface and the design of supporting documents often do not match; they may differ strongly in both content and presentation. This is caused by the separate design and production processes of user interfaces and additional help in most organizations. Because of the increasing complexity of technological products, there seems to be a growing consciousness that the user interface and the supporting instructional documentation should be considered and designed in close cooperation. Designers should realize that users may build two different types of mental models when trying to operate a device, using both the user interface and the documentation. Further integration of the design of user interfaces and user instructions may lead to a more effective and efficient use of modern technological products. If designers work together more closely, this may reduce the risk that users construct conflicting mental models of one and the same product.

1 Forms dichotomy

About a decade ago, Sony developed the *My First Sony* line of products. These were simple radios, alarms, cassette-players and other electronic consumer products. The designs expressed clearly that these products were made for children: all plastic, bright primary colors (very much like the colors of Lego bricks), rounded corners, big buttons, sturdily built, clear product graphics, and so on. This line of products was an immediate sales success¹ and the design was quickly copied by several competitors. One of these 'me-too-products' from a competitor is the *Video Painter* graphic tablet. For this product, in much the same way,

[@] p.h.westendorp@io.tudelft.nl

[@] c.jansen@let.kun.nl

[@] rob@punselie.com

¹ Children liked them, but it is noteworthy that also elderly people appeared to buy these products because of their nice big buttons, big displays and clear products graphics.

every detail expresses that this is a toy, not a tool.

The Video Painter graphic tablet can be connected to a TV and it is comparable to *Paint* software. It has quite a few functions (including some games) and - as a result - it is not so easy to use. Informal tests confirmed this hypothesis. To make optimal use of this product, most users will have to consult the manual. This is not to blame the designers of the product, it is standard for all products that have many features. But while the design of the user interface of this product so clearly suggests that it is a toy for children, the design of the user manual just as clearly suggests that it was designed for an elderly civil servant. It is a black and white document with a boring lay-out, only very few pictures, an ugly typeface and a very low print quality (probably from a low dpi dot matrix printer). Moreover, it gives the distinct impression that the text was written by a retired technician with strong interest in formal contract writing from the end of the last century. The illustrations must have been produced by a technical illustrator specializing in electronic circuit diagrams. It is an understatement that the Interface Design & the Document Design of the Video Painter do not match. Immediately after unpacking, the joy of the product is destroyed by the sadness of the manual.



Figure 1.
The Video Painter graphic tablet: a toy designed for children.

FROM THIS POINT ON, ALL DRAWING FUNCTIONS ARE ACTIVATED THE SAME AS ABOVE UNLESS OTHERWISE NOTED.

To draw straight lines:




1. Press  (Drawing Tool).
2. Choose  (THIN LINE) or  (BOLD LINE) and press the **MARK** button.
3. Locate the beginning point of the line by placing the Video Pen on the Drawing Pad. Press the **MARK** button. A small dot will appear on screen.
4. Decide on a second point and press the **MARK** button. A straight line will automatically be drawn to connect the two points.

Figure 2.

The Video Painter graphic tablet manual: a tool designed for technical professionals.

This contrast between the style of the product user interface and the style of the document interface of the Video Painter graphic tablet may be an extreme example of a mismatch between product user interface design and document design, but it is rather more the rule than the exception that the design of user interfaces of devices, and the design of documents do not match. Style is only one aspect of possible differences between the designs of each of these two kinds of products. The *design of a device interface* may refer to various aspects, such as functionality and structuring of the functions, grouping, lay-out and design of buttons, handles and display, use of color and product graphics. The *design of an information interface* also refers to various aspects, such as content and structuring of the information and overall concept of the document, lay-out of the individual pages, syntax and wording, design of the graphic elements, use of color and the typography. *Design* may involve both *content* and *presentation* and presentation may concern both *style* and *ergonomics* of the product user interface and documentation. Of course, one could argue that we are comparing incomparable products: the interface of a technical device, for instance an electronic appliance or software, and the interface of the information, usually resulting in a (printed) manual. But it is obvious that there must be a close relation in the design (as defined above) between the two. If the product interface gives the user a somewhat different idea of what the product can do and how to do that from what the documentation does, there is a mismatch between the *design* of the device interface and the *design* of the document. The differences may concern both *content* and *presentation*.

1.1 Content

At first sight, it seems obvious that at least the *content* of both must be equal: all functions that can be performed using the interface, should be covered in the manual (either on paper or online). Exceptions are possible, though. One could occasionally argue that some functions can be operated so easily that they do not have to be mentioned in the manual. This is probably a dangerous track to follow, as 'it is impossible to underestimate a vast public': there will always be users that do not see the obvious. On the other hand, computer software may have functions that are not mentioned in any manual, perhaps to invite people to try and find them. An old example is the unmentioned shortcuts in the first generation Apple Macintosh computers; there proved to be more shortcuts and functions than were mentioned in any of the documentation. John Carroll could have mentioned these as an example of the Minimal Manual principle of 'encourage active exploration' (Carroll, 1990).

1.2 Presentation

Most people will agree that the *style* of the manual should reflect the image of the product. The style is the first aspect of the design that users perceive, both on the product user interface and in the documentation. On a device, the style is defined by the shape of buttons, handles, display and other elements, use of color and product graphics. The visual style of the documentation is defined by lay-out, design of the graphic elements, use of color, typography and other aspects; the textual style is defined by syntax and wording. The example of the Video Painter mentioned above is a clear example of a visual design of a document that does not have the style that the product has². Especially products with a high brand image (which is usually reflected by the style of the user interface) need documentation that is designed with a matching style. Buyers of a Rolls-Royce automobile, a Bang & Olufson television set or a Leica camera would be unpleasantly surprised to receive a number of deteriorated black-and-white copies stapled together when they collect their new car, TV or camera from the dealer.

Perhaps most important, *ergonomic* aspects of the presentation of the device user interface and the presentation of the document interface also have to be in line with each other. Ergonomic aspects of the presentation of a device user interface concern the structuring of functions and navigation tools, grouping and lay-out of buttons, handles and display, (ergonomic) use of color and product graphics. Ergonomic aspects of a document interface presentation concern structuring of the information, navigation and overall concept of the document, lay-out of individual pages, (ergonomic) use of color and textual aspects such as syntax and wording. The need for integration and overlap in presentation of these elements is discussed by several authors contributing to this book.

² In fact, the textual style of the documentation did not match the style of the product either.

Of course, device interfaces and information interfaces differ in many respects.

Some examples may show the variety of possible differences.

- A user manual that simply describes the functions of all the menu-items one after the other is generally considered not to be a good manual, exactly because it is not user-oriented, but function-oriented (one might even say 'button-oriented'). Such a manual is easy to write, but unhelpful when the user wants to perform tasks or has a specific problem.
- Photographs may present the most exact representations of the elements, but they are generally considered not to be most efficient for users (for instance, because they show too many details); more abstract black and white line drawings are often far more useful.
- It may be useless to repeat the color-schemes used to indicate which buttons concern the same type of functionality (for instance a green background for the number-buttons on a remote control for a VCR) in the manual. On the other hand, too often this visual support of the user interface is not fully exploited in the manual.
- The use of a product may sometimes best be expressed with a manual that is basically a flowchart (see figure 3). The device interface does not necessarily have to express this flowchart concept (although it might also be an idea for some product interfaces).

The design of a device user interface and the design of documentation cannot match completely if we compare a device with a printed manual. But the designs have to overlap much more when both the user interface of the product and the interface of the information are presented on-line. Clearly, the structuring of the information must have some relationship with the structuring of the buttons or menu-items; color-schemes used in the user interface are preferably reflected in the manual or the online help, and illustrations in the manual should have a good resemblance to the depicted elements of the product interface. Navigation elements that guide users through the Help should reflect the navigation elements that guide the user through the product's functionality. Differences could be useful, though: with the ongoing integration of the Help into the functionality of the software, it might be necessary to make clear what is Help and what is not - there might be good arguments not to integrate Interface Design & Document Design completely.

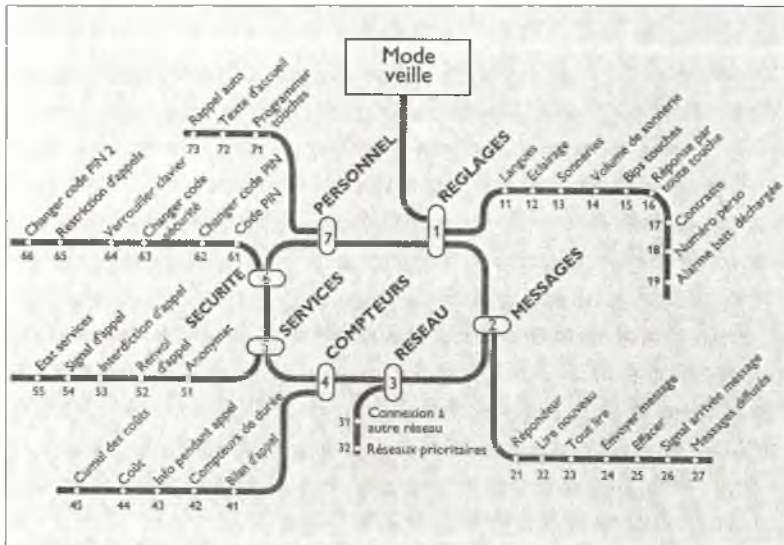


Figure 3.

Bosch manual, showing the functions of a mobile phone. It might be helpful for the product user interface to be based on the same idea as this manual is; every function immediately traceable and selectable, and a one-to-one relationship with the manual.

2 Causes

There are other reasons why product interface design and document design are usually worlds apart. In many companies, product design (which often includes user interface design) and manual production are two completely separated organizational routes; often, they are produced by two different departments. User interfaces are usually developed in close cooperation with the development and marketing of product features and documentation literally and figuratively comes last. The documentation or help is designed when the product is finished. A document designer may discover that some things are almost impossible to explain or instruct, for instance because they are very counter-intuitive, but his or her comments may come too late - and are often unwanted. Production has already started and the manuals have to be ready yesterday. There is no time to think about the design at all, let alone time to listen to the comments of the document designers and reconsider the device interface. These first 'testers of the product' are often experienced in noticing the communicative shortcomings of user interface aspects. Often, it would have been beneficial to the quality of the product, if the observations of the document designers had been taken into account.

An example of a product with an interface that makes good document design a very difficult task, is the Sony RM-816 remote control for television sets. This is a two-sided remote control that can be slid into a hard plastic holder. One side is for day-to-day use; the other side for programming and other more advanced functions³. People who want to use one of the advanced functions, press two buttons on the sides of the holder, lift the remote control out of the holder, turn it around and slide it in again until it clicks in. Then they have a wealth of buttons to choose from. But for programming, some buttons (like: 'store') that seem to be necessary are not on the remote control. Yet, the manual mentions them and shows the icons. The user may try the other side of the remote control again, but no, the buttons are not there. The user may try and reverse the remote control again, but will probably not find all of the programming buttons. It takes many users quite a while to find out that some buttons on the programming side appear to be hidden in the holder, when the remote control is slid in (see figure 4). They have to pull it out a little bit to be able to touch some of the buttons that are necessary for programming (see figure 5). A document designer may put this in the manual 'in neon lights', but many users will not notice this, simply because the design of the product showed them the wrong direction, which seemed so obvious and intuitive: for programming, use the programming side-period. A minor change in the design of the holder (e.g. 1 centimeter more open space) would have avoided the whole problem. Now, the document designer gets an impossible task, simply because there was no cooperation between the Interface Designer & the Document Designer.

In many companies that produce high-tech products, the help is considered less important because it is after sales, whereas the user interface is pre-sales, and that is where the money is. This seems a regrettably short-sighted view, possibly triggered by the urge for product managers to meet short-term targets. A product with a smart looking interface and a poor manual (be it because of content or presentation) may sell well anyway. But when the customer cannot get the newly bought microwave-oven, camera or spreadsheet with all these magnificent functions to do what it should do, the brand image that has been created with so much design, marketing and advertising money, may be devalued completely within minutes. Product managers do not always realize what this may do to the marketing of the next generation of their products and to the image of other products that are marketed by the same company. In many major companies, the relation between product's Interface Design and Document Design can be characterized as 'Penny Wise, Pound Foolish'⁴.

3 A sales assistant once called the side with the programming functions the 'daddy side' and the side with just the day-to-day functions the 'mommy side'. The (female) customer left the shop without buying the TV.

4 Schriver (1979: p. 212-223) found that 79% of the respondents in a customer survey considered the quality of the manual to be an argument to buy a product of the same make next time and that 84% thought that companies should advertise the quality of their manuals.

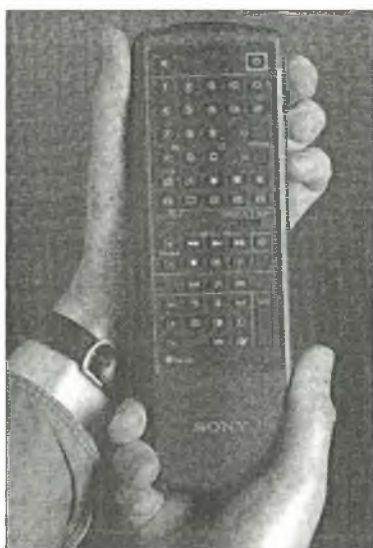


Figure 4.
The Sony RM-816 remote control;
programming side in the holder.



Figure 5.
The Sony RM-816 remote control; programming
side slightly drawn out of the holder.
Now the vital programming buttons at the
bottom are visible.

3 Turning tide

Processors and memory chips have become very powerful and very cheap and that process is still going on⁵. Electronic products now already have many functions and each new release will have many more new functions. As a result, these products are already very complex to use and will be even more difficult to use. This may cause the criterion for a marketing success to move away from technology to communication. Marketing success may no longer depend upon the number of functions, but rather on the communication of these functions. This communication will not just be the pre-sales communication, like the user interface and the marketing communication (advertising), but also the instructions for use. Right now, this may still seem wishful thinking on the part of document designers, but especially in the software sector, there are some indications of a turning tide. The most obvious reason why companies have learned to listen more to document designers is that helpdesks have proved to be necessary and expensive and that many complaints and questions could have been avoided by having a good manual⁶. Saving on the manual means spending on the helpdesk instead⁷.

⁵ Moore's Law gives an indication of the increasing performance of microchips: Gordon Moore noticed in 1964 that up to that time the performance of the microchip had doubled every year. In later years the definition has changed (with Gordon Moore's approval) to reflect that doubling occurs every 18 months.

⁶ <http://www.cyberjuice.com/helpdeskfunnies/index.htm>

⁷ See Redish, J. & Ramey, J. (1995), *Measuring the value added by professional technical communicators*. Technical Communication 42, p. 23-86.

This turning of the tide may be expressed by the results of some recent surveys concerning the use and appreciation of manuals. Schriver (1997, pp. 209-223) found that 94% of her subjects use the manual that comes with electronic products. This confirms prior research that indicated that manuals are used widely (Petersen, 1984, Wright, 1981) - in contrast to what seems to be generally believed. Schriver's research also shows an increase in the use of instructions in comparison to this prior research, and this may well be explained by the increasing complexity of the products. We simply cannot use modern (electronic) products without the manual anymore, just as we would not be able to use them without user interfaces (imagine how to operate a VCR if you would have to connect the wires yourself instead of pressing buttons).

4 Integration

The most interesting cause for a turning of the tide may be the development towards further integration of user instructions in product interfaces. In modern software, the Help already is integrated into the program. Buyers will consider the Help as belonging to the interface, comparable to the grouping of menu-items and buttons, design of direct manipulation buttons, use of color-schemes, and so on. This process will continue: the Help will increasingly be considered an essential part of the product interface. It will co-determine the user interface. An intriguing example of this integration is the World Wide Web, which started as - and to a high degree still is - a huge compilation of documents. Interface Design here means, to a large extent Document Design.

The browser (interface design) and pages (document design) could at first easily be separated as different elements. But with the use of other data formats (sound, animation, video), web-pages can hardly be called 'documents' anymore. Nowadays, plug-ins such as Java and Flash provide increasing interactivity to web documents. In fact they provide them with an interface of their own. This means that one cannot only click toward other pages with information, but that one can actually *operate* products from a distance⁸. On the web, document design thus develops into interface design⁹. With new web standards - such as XML - this trend will only increase. Now, more and more, the interface design becomes dominant: not just the design of the navigation and the selection of information on pages, but the design of an interface that controls a product, whether elements on the website itself or other software or indeed hardware (such as a video camera). Moreover, the

8 Nice examples concerning the integration of Interface Design & Document Design can be seen and used at <http://www.livemanuals.com> and at <http://207.97.150.242/instructions/animations.html>

9 An interesting reverse angle can be found in software agents (or 'intelligent agents'), small pieces of software that assist users by autonomously performing specific tasks on their behalf, such as information retrieval or price negotiations. This could be interpreted as software (including an interface) focusing on document design related issues. <http://www.cis.udel.edu/~agents98/> <http://pattie.www.media.mit.edu/people/pattie/>
For more links to intelligent agents, see the references in José Arcellana's *Designing Assistants* in this volume.

combination of navigation through information and controlling of remote software or hardware, plus instructions to do so, will make websites much more complex than they are nowadays and make Help more vital.

This development of complete integration of product interface and instructions for use has only just started. Until recently, help was provided online by the program, but users not only had to find the information they needed themselves, they also had to feed this information back into the computer that obviously already knows this information. Probably version 1 of WordPerfect for the Macintosh was the first software with integrated interactive help. Once the user had selected a subject, the program not only offered the procedural information, but also an 'Execute' button in the Help with which users could immediately perform the action (see figure 6). The instructions were presented, such that the user could enter them in the computer manually, but the computer could also perform the task if the user just hit the 'Execute' button. As a result, the user did not have to learn how to perform this task anymore. This means that the online-help in fact provided the user with an alternative user interface for performing the task. Theoretically, in such a situation (if all functions could have been operated in this way), the user could perform all tasks without using the 'standard' user interface (menus, menu-items); he or she could always simply immediately select the Help, find the function and press the 'Execute' button. It may be a bit of an exaggeration because it seems impossible to apply this 'Execute' button for all tasks, but one could suggest to completely replace the product user interface (consisting of menus, menu-items, and so on) with just the Help interface.

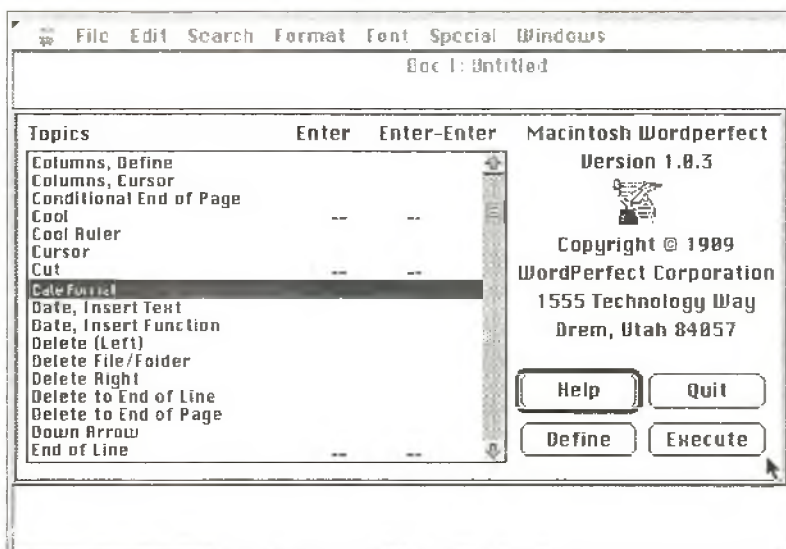


Figure 6.
Help in WordPerfect for the Macintosh version 1.05. Notice the 'Execute' button.

Remarkably, WordPerfect discontinued this functionality in the Help, but Microsoft introduced it in the Office 95 Help and further developed it in Office 97 and Office 2000. In MS Office, the user asking for help not only gets information, but may also get a wizard that guides him or her in performing the task immediately from within the help function. The contribution of Hoek & Kaufman in this book presents an excellent overview of this development and makes clear that Microsoft intends to proceed along this line of integrating Interface Design & Document Design.

5 Mental representations

Users perceive and learn the functions of a product in various ways:

1. by operating the product user interface (icons, buttons, and so on)
2. by consulting the help (printed manual or the basic online help)
3. via all other kinds of instructional information (error messages, wizards, agents, context sensitive help, cue cards, coaches, demos, tutorials, classroom trainings).

By using the interface, by consulting the help and by perceiving all the other information, users develop a kind of mental model (for technical products sometimes termed a *device model*) of that product: a little theory of how this product functions or should be operated. This device model will be adapted or refined during use of the product and may always change.

Moreover (4), users may already have a simple form of a device model even before they have used the product. This elementary device model has been developed on the basis of all kinds of pre-sales information, both commercial and non-commercial. Users may have seen advertising (commercials, brochures) or received other pre-sales information (from the reseller or from the neighbor). The device model will also be developed on the basis of association of the new product with products that they know¹⁰. A user who buys a new VCR will develop a mental model of this VCR on the basis of the ones he or she already used. In the case of a complete new type of product, the user will develop a mental model on the basis of a comparison with existing products that have similar functionality. For instance, a user who has just bought new hardware and software to record broadcasted programs onto a hard disk, may *a priori* have the idea that such recording will be similar to that with a VCR, because a VCR has a comparable main function: recording and reproducing broadcasted programs. For an overview of the development of the user's device model see figure 7.

¹⁰ Hackos & Redish (1998, p. 408) present a nice diagrammatic overview of all the communication tools that provide information or instruction for users of technical products. This diagram also shows the proximity of the various tools related to the product and the user. Unfortunately, Hackos & Redish do not include the pre-sales information.

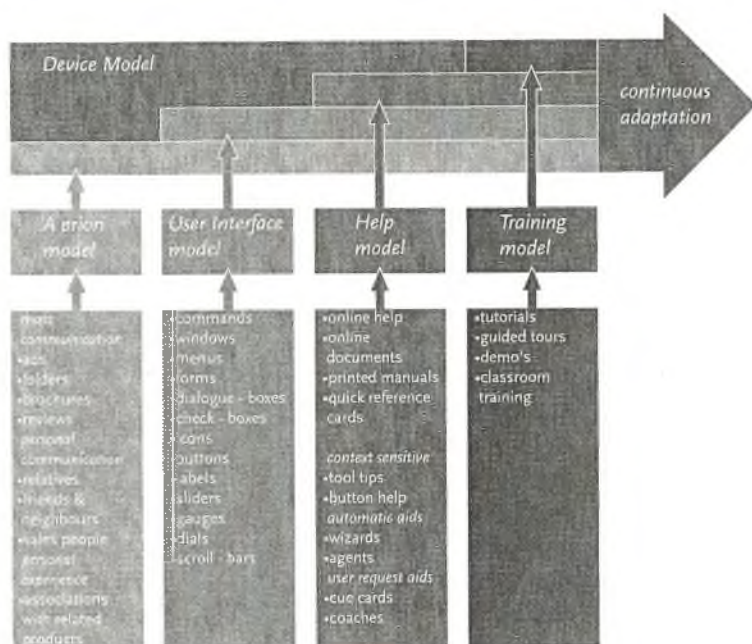


Figure 7.

Development of the user's device model on the basis of various sources of information: a priori information, user interface, help and training aids. This diagram shows the combination of influences on the user's device model, based on both User Interface and Documentation aspects. The various sub-models may interfere or even conflict with each other. Note that some items could be classified as both User Interface and Help elements (e.g. tool tips, button help, wizards, agents, cue cards and coaches).

This diagram also expresses increasing user initiative in searching for information over time - from passively receiving information (a priori) to actively searching for information (training aids).

An interesting question is whether the user of a product that comes with a manual and/or online help, builds up just one device model, based on the user interface with support from the manual, or two different device models (one based on the visual interface and one based on the instructions in the manual) or two different ones, depending on the degree to which additional help was used. The device model that a user builds up or develops by just trying out all buttons might well be different from the device model that a user builds up by consulting the manual or online help. The latter activity implies following a different learning track and, in doing this, the user may perceive the functionality of the product differently, even if he or she has used the same functions of the product (see figures 8 & 9). It is conceivable that the user who learns just by trying out the product user interface (control panel), develops a mental representation of the product that is first of all based on the location and recognition of the elements, whereas the user who consults the manual for (and before) each task, may develop a model based on procedural instructions. The first user might build up a model that consists of an overall idea of lay-out and form; the second might build up a model that basically consists of a series of procedures.

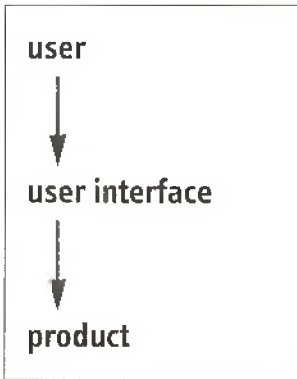


Figure 8.
User learns just by using the product
user interface.

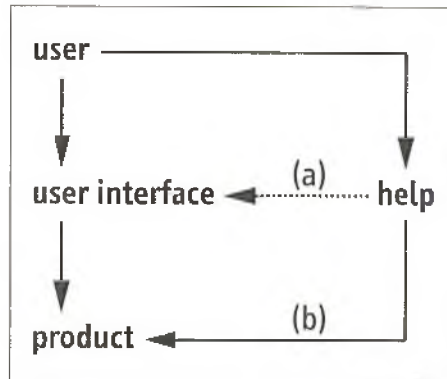


Figure 9.
User learns by using product user interface plus help (a)
or by purely using the (interactive) help interface (b).

Often, this difference will also be reflected in the opposition of pictorial elements (on-screen buttons, and so on) and textual instructions. One could hypothesize that instructions would be more efficient and effective if they would be, as much as possible, a one-to-one copy of the visual product user interface elements (icons, buttons, and so on). The cognitive load would then be minimal, because users perceive the instruction in exactly the same form as the objects that they have to manipulate. No media transition would be necessary - and there is some evidence that reproduction of instructions in another medium than they were presented in (i.e. reproduction of an instruction in text if it was offered in pictures or vice versa) is an extra burden on the cognitive load (Seel & Strittmatter, 1989).

If help functions were further integrated into the product user interface, especially if they were completely interactive (meaning that one could perform all tasks through the online help), the user would really have two alternative interfaces, both presented on the screen and in almost identical form (icons, font types, buttons). This might indeed be confusing to some users. Hoek & Kaufman mention this possible 'risk of making it difficult for users to build up a mental model' in their contribution to this volume.

6 Developments

In spite of the risk of confusing users who might develop different mental representations of the same product, user interface designers seem to be convinced that the increasing complexity of new products can only be encountered by further integration of the online help into the product user interface. In their contributions to this volume Hoek & Kaufman from Microsoft and Arcellana & Knabe from Apple Computer clearly represent this point of view.

Hoek & Kaufman notice that 'the effort of using on-line help may badly interrupt the user's workflow'. They describe the efforts that Microsoft has made - and to which they have contributed - to further integrate the help into the product user interface and the problems that accompany such a development. Their article provides an intriguing view of the problems that arise when trying to integrate the help into the product user interface. Hoek & Kaufman show how they try to let the users concentrate on their work, also when they need help. Hoek & Kaufman present the help in such a way that users do not have to switch back and forth between the product interface and the help interface¹¹.

Knabe describes how Apple Computer has worked towards integration of the help into the product user interface on the basis of John Carroll's minimal manual guidelines, which are based on the learning-by-doing approach. Users should always be encouraged to try out new functions without being afraid of losing track, losing time or losing work. Knabe shows how the Apple Help evolved from answering the 'How do I..?' questions to 'Why don't you..?' Help when the user needs it is one thing; trying to let the user really profit from all functions that may be of use is quite another. Knabe takes the problem that Hoek & Kaufman tackled (integrating the Help in such a way that users can focus on their work, even when they need an answer to a specific question) one step further. He tries to let the users proceed with their work as much as possible and yet makes them interested in discovering new functions.

Arcellana describes the development of the 'Assistants' at Apple Computer as an addition to the 'ordinary' Help (based on Index, Search and Table of Contents). This additional help interface is not so much directed at solving problems presented by the computer, but rather at real-world tasks that the users may have. 'The Assistants were designed to address the user's needs for support of higher level, real world goals, as opposed to lower level, computer based tasks and procedures'. Arcellana also discusses the opportunities to further seduce people into using both the standard and the additional help systems, apart from just trying out the product user interface. He suggests making interfaces for standard and additional help both visually and emotionally more attractive and discusses some ways of doing so.

Especially the elderly users of high-tech products are often reported to have problems with product user interfaces *and* with the additional help¹². Therefore, it is interesting to focus on this group of users when discussing the integration of help into the product user interface. Would further integration really help the elderly user, or would the possible disadvantages (about which, for instance, Hoek & Kaufman warn) confuse rather than help

11 This switching has been described in Jansen & Steehouder (1992) for users who have to fill in forms and by Van Hees (1996) for users of consumer products with manuals.

12 Schriver (1997: p. 507) reports a study of Kelley & Charness (1995) in which they conclude that older adults experience significantly more difficulty in learning to use a computer than younger adults.

the elderly users? Elderly users have been reported to be more careful readers of the additional help than younger users. This may be true for printed manuals, but it remains unclear whether this is also true for online help, as the screen may cause additional problems and because one does have to be somewhat computer-literate to operate an online help facility in the first place.

Bouma describes in minute detail what limitations elderly users may experience - and what should not be considered as limitations of elderly users if the environment were designed adequately. He discusses a wealth of research focussing on the development of suitable interfaces and documents for the elderly. *Bouma* argues that adaptive user interfaces, based on agent technology, can be especially helpful for this group of users. He stresses the need for designers of both product interfaces and documentation to take elderly consumers more seriously, and to undertake a concerted effort to solve their specific problems in dealing with new technology.

Wright discusses the documentation needs of older people within the context of (1) the evolution in the ways computer users are provided with information and (2) the qualitative changes that have occurred in documentation design. She specifically refers to the question whether all kinds of documents that we have today will still be needed in the future. But as long as additional documentation is still necessary or at least useful, there are three types of document design implications concerning age-associated impairments that have to be taken into account: design implications based on physical needs (sensory limitation), those based on cognitive needs (constraints on mental resources) and those based on emotional needs (personal goals). *Wright* emphasizes the need for adaptive user interfaces (from which not only elderly users can profit). She also discusses the interdependence of decisions about documentation and the user interface. She stresses that decisions concerning the design of the user interface of a product should not be taken without considering the design of the interface of the documentation and vice versa. This leads to a discussion of the importance of performance-based evaluation during the design process, rather than separate tests for the product user interface and for the documentation¹³. *Wright* points out that age is not always the most suitable variable for matching between the volunteers assisting with pilot evaluation and the target audience. She stresses that comparable knowledge, experience or interests may be more critical in some contexts.

But why have documentation at all? The need for alternatives, as mentioned by several authors contributing to this book, has been questioned and answered from a historical point of view by *Bouwhuis*. He points out that the reason for being of product

13 This conclusion from a researcher must be very valuable for many companies that produce high-tech products: not just the design of the product user interface and document interface seem to be separate tracks in most companies, but the testing of each of these just as well.

documentation can be traced back to perceived weaknesses of spoken communication. Bouwhuis argues that uncertainty is the central problem of users who try to operate products that are new to them using a manual or other kind of help. He also concludes that improvement of this communication process has to be based on a higher degree of integration between the user interface and the documentation.

Problems caused by technology can only be solved by the development of new technology. Problems for users in the operation of new devices, caused by the complexity of all the new features, can only be solved by the development of new and more integrated user interface designs and document designs. The contributions in this volume may be helpful in exploring opportunities for further cooperation among interface designers, document designers and researchers in both fields.

References

- Carroll, J.M. (1990), *The Nurnberg funnel: Designing minimalist instruction for practical computer skill*. Cambridge, MIT Press.
- Hackos, J.T. & Redish, J.C. (1998), *User and task analysis for interface design*. New York, John Wiley & Sons.
- Hees, M.M.W. van (1996), *User instructions for the elderly: what literature tells us*. Journal of technical writing and communication 26, p. 521-536.
- Jansen, C.J.M. & Steehouder, M.F. (1992), *Forms as a source of communication problems*. Journal of technical writing and communication 22, p. 179-194.
- Petersen, D. (1984), *Die Gebrauchsanweisung als kommunikatives Mittel zur Beeinflussung des Gerätebenutzers Verhaltens* (in German). [The manuals as a means of communication to influence the people's behavior when using products]. Hamburg, Doctoral Thesis.
- Schriver, K.A. (1997): *Dynamics in Document Design*. New York, John Wiley & Sons.
- Seel, N.M. & Strittmatter, P. (1989), *Presentation of information by media and its effect on mental models*. In: H. Mandl, & J.R. Levin, (eds.), *Knowledge acquisition from text and pictures*. Amsterdam, Elsevier Science Publisher, p. 37-58.
- Wright, P. (1981), *The instructions clearly state... Can't people read?* Applied Ergonomics 12, p. 131-141.

About the authors

Piet Westendorp is a researcher at both Eindhoven University of Technology and Delft University of Technology. He has published several books and many articles concerning instructions for use. Moreover, he has worked as a freelance journalist, technical communicator and advisor for many major companies. His most recent book is *Open here: the art of instructional design* (together with Paul Mijksenaar; Thames & Hudson, 1999).

Carel Jansen holds the chair of Business Communication at Nijmegen University. He has published books and papers about optimizing the order of instructions, intercultural aspects of technical communication, and the design of instructional documents, specifically government forms and software manuals.

Rob Punselie is a creative consultant for major commercial and nonprofit organizations. He specializes in content and writing for the web and has published frequently on this subject, including his book *Wijzer op het web* ('Wiser on the web', Samsom, 1999). He is also President of the Netherlands Society for Technical Information and Communication (STIC).